spi_psd_si

User Manual

Version 1.4.2
5 February 2003

Jürgen Knödlseder
Centre d'Etude Spatiale des Rayonnements
knodlseder@cesr.fr
http://www.cesr.fr./~jurgen/index.html

Note to the user

This software has been written to analyse data of the SPI telescope onboard INTEGRAL. Particular care has been taken in making the software user friendly and well documented. If you appreciated this effort, and if this software and User Manual were useful for your scientific work, the author would appreciate a corresponding acknowledgment in your published work.

Contents

1 Introduction

The executable `spi_psd_si` estimates the PSD sensitivity improvement as function of energy from an input group (science window group, observation group, or index group) for a user defined time window. For this purpose, `spi_psd_si` determines the significance of gamma-ray line detections with and without taking into account PSD information.

The significance of a gamma-ray line detection is estimated separately for each of the 19 SPI detectors by fitting Gaussian shaped line profiles on top of a polynomial to the detector spectra. For each fit, the energy of the requested gamma-ray line as well as an energy window used for fitting has to be specified. Within this energy window, multiple gamma-ray lines may be present, although the most reliable results are probably achieved for isolated gamma-ray lines where the background spectral shape can be securely determined from adjacent energies. `spi_psd_si` automatically detects such *secondary* gamma-ray lines in the spectra and adds additional Gaussians to the fit function in order to account for them properly.

`spi_psd_si` should be applied to sufficienctly strong instrumental gamma-ray lines that are based on real photon interactions throughout the PSD energy range (∼200 keV - 2.3 MeV). Such a tentative list based on the analysis of calibration data is compiled in section 6, and may be used as default input list to `spi_psd_si`. Note that the instrumental lines should at least achieve a detection significance of $\sim 10\sigma$ in data (**TBC**) that do not take into account PSD information to make the method work. Higher significances, of course, improve the reliability of the results.

`spi_psd_si` has been designed to execute in any kind of pipeline, such as the science window pipeline or the revolution pipeline (however it could also be applied to an observation group for deep performance analysis). Since a sufficient number of PSD events is required for reliable PSD sensitivity improvement estimation, the **revolution pipeline is the priviledged location for** `spi_psd_si`. `spi_psd_si` has been designed to append regularily sensitivity improvement estimates to a file, and hence to build a time history of the sensitivity improvements. To make the logic work, `spi_psd_si` needs an index group on input, which it will search automatically for sufficient **new data** (i.e. data that are dated after the last entry in result file) for sensitivity improvement computation. If not enough new data is available, `spi_psd_si` will do nothing.

`spi_psd_si` is written in the ANSI C++ language. The task has been developped under ISDC support platform 5.2 and requires `spi_psdlib` version 2.0.1 and `spi_toolslib` version 2.0.2 or higher.

2 Getting started

Before installing `spi_psd_si`, make sure that the ISDC support platform 5.2or higher is installed on your system, and that you have installed the libraries `spi_psdlib` version 2.0.1 and `spi_toolslib` version 2.0.2 or higher.

After downloading the `spi_psd_si.tar.gz` file, step into a directory that should hold the distribution, move the `spi_psd_si.tar.gz` file into this directory and type:

```
$ gunzip spi_psd_si.tar.gz
$ tar xvf spi_psd_si.tar
```

The first command uncompresses the distribution file, the second unpacks the files.

Before configuration, the distribution needs to be reset to a clean state. To do this, type

```
$ make distclean
```

Then, configure the distribution. It is assumed here that you have previously installed the ISDC support platform, thus you should type

```
$ ~/bin/ac_stuff/configure
```

Finally, build the distribution by typing

```
$ make global_install
```

To perform a unit test, type

```
$ make test
```

Make sure that the test data `spi_test_data-1.0.tar.gz` are available at your site (they should reside in a directory whose name is defined by the `ISDC_TEST_DATA_DIR` environment variable).

3 Parameter file

```
##############################################################################
#                                                                            #
#               Centre d'Etude Spatiale des Rayonnements                     #
#                     (in collaboration with ISDC)                           #
#                                                                            #
#              SPI PSD sensitivity improvement estimation                    #
#                                                                            #
# ---------------------------------------------------------------------- #
#                                                                            #
#  File:      spi_psd_si.par                                                 #
#  Version:   1.4.2                                                          #
#  Component: osm                                                            #
#                                                                            #
#  Author:    Juergen Knoedlseder                                            #
#             knodlseder@cesr.fr                                             #
#             CESR                                                           #
#                                                                            #
#  Purpose:   Parameter file of the SPI PSD sensitivity improvement          #
#             estimation executable                                          #
#                                                                            #
#  History:   1.4.2   5-Feb-2003  First ISDC delivery (Rev. 4)               #
#                                                                            #
##############################################################################
#
# Input DOLs
#===========
inDOL,    s, ql, "swg_prp_idx.fits[1]",,,"Input Group DOL (SWG/OG/IDX)"
lineDOL,  s, ql, "psd_si_lines.fits[SPI.-LINE-SCT]",,,"Line definition DOL"
coeffDOL, s, ql, "spi_gain_coeff_idx.fits[1]",,,"Gain correction (File/IDX)"
alertDOL, s, ql, "psd_limits_idx.fits[1]",,,"Alert Limit DOL (File/IDX)"
#
# Output DOL
#===========
outDOL, s, ql, "si.fits",,,"Output DOL (HDU optional)"
#
# OBT limits
#===========
minOBT, s, ql,      "",,,"Event usage minimum OBT"
maxOBT, s, ql,      "",,,"Event usage maximum OBT"
append, b, ql,     yes,,,"Append minimum OBT to last results ?"
slice,  b, ql,     yes,,,"Split time interval in constant ONTIME intervals?"
nopart, b, ql,     yes,,,"Skip partial time intervals ?"
ontime, r, ql, 10000.0,,,"Constant ONTIME slice (seconds)"
#
# SPI mode and PSD discrimination
#================================
onground,  b, h,  no,,,         "Use onground discrimination ?"
siThres,   r, h, 4.0, 2.0, 99.0,"Required line detection significance"
engtolRel, r, h, 1.0, 0.0, 99.0,"Required relative energy tolerance (keV)"
#
# Sensitivity improvement fit parameters
```

```
#=====================================
ignore,   b, h, yes,    ,     , "Ignore fit alerts ?"
bgddop,   i, h,   1,   0,    3, "Degree of background polynomial"
maxlines, i, h,   1,   1,    3, "Maximum number of lines fit simultaneously"
thres,    r, h, 4.0, 2.0, 99.0, "Line search threshold (sigma)"
engtol,   r, h, 5.0, 0.0, 99.0, "Primary line energy tolerance (keV)"
engtol2,  r, h, 5.0, 0.0, 99.0, "Secondary lines energy tolerance (keV)"
minSigma, r, h, 0.5, 0.5, 10.0, "Minimum line width sigma (keV)"
maxSigma, r, h, 2.0, 0.5, 10.0, "Maximum line width sigma (keV)"
#
# Diagnostic parameters
#======================
reportpar, b, h, no,,,"Report line fitting parameters ?"
reportfit, b, h, no,,,"Report line fitting details ?"
#
# Limit checking definitions
#===========================
limcheck, b, h,   yes,,, "Perform limit checking ?"
alert0,   b, h,   yes,,, "Generate level 0 alerts ?"
alert1,   b, h,   yes,,, "Generate level 1 alerts ?"
alert2,   b, h,   yes,,, "Generate level 2 alerts ?"
alert3,   b, h,   yes,,, "Generate level 3 alerts ?"
minPE,    i, h,  1000,,, "Minimum number of PE for limit checking"
#
# ISDC Standard Parameters
#=========================
clobber, b, h, no,,,   "Overwrite existing data structures ?"
mode,    s, h, "ql",,, "Execution mode"
```

The following parameters have to be specified:

- `inDOL` specifies the input DOL (science window group, observation group, or index group) for which the PSD sensitivity improvement should be derived. A ISDC level of `PRP` is sufficient if a calibration coefficient file or index file is specified in the `coeffDOL` parameter. Energy correction is in this case performed *on-the-fly*. However, if the input group is of level `COR`, the calibrated energy information from the corresponding data structures is taken, irrespectively of the presence of a calibration DOL. Note that if `onground` pulse shape analysis errors should be considered, the `COR` level is needed in any case!

- `lineDOL` specifies the line definition DOL (of HDU type `SPI.-LINE-SCT`) that defines which lines should be fitted by `spi_psd_si`.

- `coeffDOL` specifies a calibration file (`SPI.-COEF-CAL`) or calibration index file (`SPI.-COEF-CAL-IDX`) that is used for energy calibration **if the ISDC level of the input group is** `PRP`.

- `alertDOL` (optional) if alert limit checking is requested (`limcheck = yes`), this parameter specifies the DOL of the alert limit file `[SPI.-ALRT-LIM]` or the alert limit index `[SPI.-ALRT-LIM-IDX]` (including the HDU).

- `outDOL` specifies the output **filename** into which the sensitivity improvement estimations are written. The specification of the HDU `[SPI.-SIGN-PSD]` is optional, but not required by the task. If the data structure exists already, `spi_psd_si` appends rows to the existing table. If the data structure or the file does not exist, `spi_psd_si` creates a new file/HDU.

- `minOBT` specifies the minimum OBT limit of the events that should be used for sensitivity improvement determination. The OBT format is a character string. Leading `0` may be omitted. If the character string is empty, or if any non-number character is specified (such as "no" for example), no mimimum OBT limit is applied (and data accumulation starts with the first event in the input group).

- `maxOBT` specifies the maximum OBT limit of the events that should be used for sensitivity improvement determination. The OBT format is a character string. Leading `0` may be omitted. If the character string is empty, or if any non-number character is specified (such as "no" for example), no maximum OBT limit is applied (and data accumulation stops with the last event in the input group).

- `append` specifies if the minimum OBT limit should be set to the last OBT that occurs in the output file (specified by `outDOL`) in order to produce a continuous set of PSD sensitivity improvement estimates. The last OBT will be extracted from the keyword `OBTLAST` in the output file. **This parameter is only active if `minOBT` has not been set by the user**, i.e. `minOBT` has precedence and will not be overwritten.

- `slice` specifies if the input group should be "sliced" into time frames of constant `ONTIME` (the `ONTIME` is the time, specified in seconds, during which SPI science data were accumulated and made available to the observer).

- `nopart` (optional) if `slice = yes`, specifies if partial time slices, i.e. time slices with durations that are shorter than the requested `ONTIME`, should be skipped. Partial time slices may occur at the end of a data stream, and to assure a uniform quality of the PSD sensitivity improvement it is recommended to set this parameter to `yes`. Together with `append = yes`, re-execution of `spi_psd_si` at a later time will append new time slices that start with the OBT of the last appended time slice.

- `ontime` (optional) if `slice = yes`, specifies the `ONTIME` duration of each time slice. Note that the last time slice has generally an effective `ONTIME` that is shorter than the specified value, since in general, the available `ONTIME` is not an integer multiple of the value specified by `ontime`.

- `onground` specifies if onground or onboard pulse shape discrimination should be considered for the sensitivity improvement estimate. Since onground determination should be in general superior to onboard discrimination (due to the more universal discrimination scheme), onground PSD discrimination should be the default for scientific data analysis. Hence, as default use `onground = yes`.

- `siThres` specifies the required minimum line detection significance in all three spectra (`ALL`, `MUL`, and `SGL`) to perform a sensitivity improvement calculation. If this detection significance is not reached, the corresponding `PSD_SI` element if the `SPI.-SIGN-PSD` table will be set to 0. If `ignore = no`, `spi_psd_si` will return an `SPI_PSD_SI_ERROR_LINE_NOT_FOUND` error if the specified significance was not reached in all 3 spectra.

- `engtolRel` specifies the required maximum line energy difference among the three spectra (`ALL`, `MUL`, and `SGL`). This parameter helps to reject cases where the line fits were captured by different, in general spurious lines (or statistical fluctuations). Set this parameter to a value that is comparable with the instrumental energy resolution (e.g. 1 keV). The units of `engtolRel` is keV. If `ignore = no`, `spi_psd_si` will return an `SPI_PSD_SI_ERROR_INCOMPATIBLE_ENERGY` error if incompatible line energies have been found.

- `ignore` specifies if alerts should be ignored. In this case, corresponding entries in the `SPI.-SIGN-PSD` table will be set to `0`, yet the program execution is not stopped. As default, specify `yes`, if using `spi_psd_si` in a pipeline (this avoids task errors).

- `bgddop` specifies the degree of the polynomial that is used to fit the spectral distribution of the background events. In general, only a small energy window enclosing the gamma-ray line is fitted, hence the spectral distribution is quite well explained by a low-order polynomial.

- `maxlines` specifies the maximum number of gamma-ray lines that should be fitted simultaneously within the specified energy window. In general, only isolated lines should be used for sensitivity improvement determination, in which case this parameter could be set to `1`. However, there may be cases where no sufficiently isolated gamma-ray lines are avialable, and only multi-line fits provide acceptable results (these additional gamma-ray lines are called *secondary* lines).

- `thres` specifies the detection threshold (in Gaussian σ) for the automatic line search (see the description of the `SPISpec` function `searchLines` in the `spi_toolslib` User Manual for more information about the line search algorithm). Typically, a detection threshold of 4σ should be used to avoid spurious line detections. If `ignore = no`, `spi_psd_si` will return an `SPI_PSD_SI_ERROR_LINE_NOT_FOUND` error if the threshold has not been exceeded for all 3 spectra.

- `engtol` specifies the allowed energy tolerance for the *primary* gamma-ray line, i.e. the gamma-ray line for which the detection significance should be determined. Energy tolerance means here the maximum energy difference between the user specified line energy (see the parameters `leng`*nn* below) and the detected line energy. The *primary* line will not be searched for outside this energy interval. The units of `engtol` is keV. This parameter should be of the order of magnitude of the **absolute** calibration uncertainty. If `ignore = no`, `spi_psd_si` will return an `SPI_PSD_SI_ERROR_ENERGY_NOT_FOUND` error if the best fitting line energy is outside the tolerance limit for at least one of the 3 spectra.

- `engtol2` specifies the allowed energy tolerance for the *secondary* gamma-ray lines, i.e. the supplementary gamma-ray lines that may have been detected by `spi_psd_si` within the specified energy window in addition to the *primary* gamma-ray line. Energy tolerance means in this case the maximum energy difference between the line energy that has been detected by the `spi_psd_si` line search algorithm and the best fitting line energy (see the description of the `SPISpec` function `searchLines` in the `spi_toolslib` User Manual for more information about the line search algorithm). This parameter may help to stabilise the fit in the case of nearby or even overlapping gamma-ray lines. The units of `engtol2` is keV.

- `minSigma` specifies the minimum allowable gamma-ray line width in Gaussian σ. Since gamma-ray lines cannot be more narrow than the detector energy resolution, a value of $\sim$ 0.5 keV seems a reasonable lower limit throughout the entire energy range. Limiting the possible range of line widths will help to stabilise the fit. The units of `minSigma` is keV.

- `maxSigma` specifies the maximum allowable gamma-ray line width in Gaussian σ. In most cases, the gamma-ray lines of intrest are intrinsically narrow, hence a value slightly larger than the detector energy resolution should provide a reasonable value (as default, 2 keV is proposed). Limiting the possible range of line widths will help to stabilise the fit. The units of `maxSigma` is keV.

- `reportpar` specifies if initial and result fit parameters should be reported into the log file. As default, `reportpar = no`, yet in case of trouble with fitting gamma-ray lines, one may switch to `yes` for diagnostics.

- `reportfit` specifies if detailed fit results should be reported into the log file. As default, `reportfit = no`, yet in case of trouble with fitting the gamma-ray lines, one may switch to `yes` for diagnostics.

- `limcheck` specifies if alert limit checking should be performed by `spi_psd_si`. If set to `yes`, `spi_psd_si` compares the sensitivity improvements to the limits that are specified in the alert limit file (see `alertDOL`) and (optionally) creates ISDC alerts (see parameters `alert0` to `alert3`). Alert limit checking will be only performed for time slices that show sufficient event statistics. The event statistics limit is defined by the parameter `minPE`.

- `alert0` (optional) if alert limit checking is enabled (`limcheck = yes`), generates level 0 ISDC alerts.

- `alert1` (optional) if alert limit checking is enabled (`limcheck = yes`), generates level 1 ISDC alerts.

- `alert2` (optional) if alert limit checking is enabled (`limcheck = yes`), generates level 2 ISDC alerts.

- `alert3` (optional) if alert limit checking is enabled (`limcheck = yes`), generates level 3 ISDC alerts.
- `minPE` (optional) if alert limit checking is enabled (`limcheck = yes`), specifies the minimum required number of PSD event (PE) to initiate alert limit checking. This parameter avoids alert limit checking in case of insufficient event statistics. To achieve reasonably good PSD sensitivity improvements, a minimum number of **TBD** PSD events (for each detector) should be requested.
- `clobber` ISDC standard parameter (not used so far).
- `mode` ISDC standard parameter (not used so far).

4 Interface definition

`spi_psd_si` derives the PSD sensitivity improvement from a stream of photon data. This photon data may be either grouped in a science window group, an observation group, or an index file. `spi_psd_si` needs at least an ISDC level of `PRP`. In this case, a calibration file (`SPI.-COEF-CAL`) or calibration index file (`SPI.-COEF-CAL-IDX`) has to be specified, and energy calibration is performed *on-the-fly*. However, if the input group is of level `COR`, the calibrated energy information from the corresponding data structures is taken, irrespectively of the presence of a calibration DOL.

If onground analysis errors should be considered correctly for efficiency and threshold calculation, the `PSD_CORFLAG` column of PSD events (`PE`) has also to be filled, and an ISDC level of `COR` is now mandatory. Make also sure that `spi_psd_postprocess` has been executed before calling `spi_psd_si`.

`spi_psd_si` scans the input group for all single event (`SE`) and PSD event (`PE`) data it can find and builds three spectra:

- one where `SE` and `PE` are simply added together (the *total spectrum*)
- one where only `PE` multiple-site events are put into a spectrum (the *PSD spectrum*)
- one where `SE` and `PE` non-multiple-site events are added together (the *not PSD spectrum*)

An OBT interval may be specified that selects a particular sub-interval for which the sensitivity improvement should be determined. In addition, one may slice the specified OBT interval in time intervals of constant `ONTIME` (this option guarantees a constant gamma-ray line detection significance for all time intervals).

For each time-interval and gamma-ray line, `spi_psd_si` adds one row to the `SPI.-SIGN-PSD` result data structure. If there are `n` gamma-ray lines for which the sensitivity improvement should be estimated, `spi_psd_si` adds blocks of `n` rows with identical OBT interval and `ONTIME` but different gamma-ray line energies. `spi_psd_si` fills all columns of the `SPI.-SIGN-PSD` data structure, hence it can be considered as complete after the task has finished. The `OBTFIRST` and `OBTLAST` keywords are also updated, so that index group generating tools may be used to assess the validity interval of the data structure. In particular, `spi_psd_si` may access the `OBTLAST` keyword set by a previous run if continuous time slices should be added to the output file (parameter `append = yes`). The following columns are filled by `spi_psd_si`:

- `OBT_START` : OBT start of the time interval of the actual row.
- `OBT_STOP` : OBT stop (or end) of the time interval of the actual row.
- `ONTIME` : ontime for this row in seconds.
- `LINE_ENERGY` : gamma-ray line energy for which the sensitivity improvement has been estimated (the values specified by the task parameters `leng`*nn*.

- `LINE_SIG_ALL` : the gamma-ray line detection significance for all 19 detectors without considering PSD information. A value of `0` indicates that either the corresponding fit failed, or the best fitted line energy was outside the specified energy tolerance (see parameter `engtol`), or that no line has been found above the threshold `thres` by the line search algorithm.
- `LINE_SIG_MUL` : the gamma-ray line detection significance for all 19 detectors under consideration of only PSD multiple-site events. A value of `0` indicates that either the corresponding fit failed, or the best fitted line energy was outside the specified energy tolerance (see parameter `engtol`), or that no line has been found above the threshold `thres` by the line search algorithm.
- `LINE_SIG_SGL` : the gamma-ray line detection significance for all 19 detectors under consideration of only single-detector and PSD non multiple-site events (single-site events and analysis error events). A value of `0` indicates that either the corresponding fit failed, or the best fitted line energy was outside the specified energy tolerance (see parameter `engtol`), or that no line has been found above the threshold `thres` by the line search algorithm.
- `PSD_SI` : resulting sensitivity improvement for all 19 detectors. A value of `0` indicates that either the `ALL` or both the `MUL` and `SGL` fits failed, that either the `ALL` or both the `MUL` and `SGL` fits resulted in a line energy that is incompatible with the energy tolerance `engtol`, or that both the best fitting `MUL` and `SGL` energies were incompatible with the best fitting `ALL` energy, or that the sifnificance of either the `ALL` or the combined `MUL` and `SGL` fitting results was below the threshold `siThres`.
- `SE_NUM` : number of single detector events (`SE`) that have been collected for sensitivity improvement determination.
- `PE_NUM` : number of PSD events (`PE`) that have been collected for sensitivity improvement determination.
- `PE_NUM_SGLE` : number of PSD events without pulse shape analysis errors that have a single-site flag (i.e. background events).
- `PE_NUM_MULT` : number of PSD events without pulse shape analysis errors that have a multiple-site flag (i.e. photon events).

Optionally, `spi_psd_si` is able to generate ISDC alerts if some of the sensitivity improvement parameters fall out of the defined limits. The alert limits are defined by an `SPI.-ALRT-LIM` structure from which the following columns are used by `spi_psd_si` for alert generation:

- `PAR_NAME` specifies the parameter for which the limits apply. Valid parameter names are `LINE_SIG_ALL`, `LINE_SIG_MUL`, `LINE_SIG_SGL`, and `PSD_SI`. If one of those parameter names is defined, the specified limits apply to **all** 19 PSD channels. By adding `_L`*n* to the parameter name (where *n* runs from 0 to 18), limits may be specified for a given PSD channel (for example `PSD_SI_L3` specifies the limits for the PSD sensitivity improvement of channel 3). Channel specific parameters have precedence over common parameters (i.e. those without the `_L`*n* extension), hence one may define common limits for all 19 PSD channel and overwrite a few limits for specific channels by specifying explicitely the channel number.
- `MIN_VAL` specifies the lower limits (inclusive) for the four ISDC alert levels (DAL table columns 1-3 corresponds to alert levels 0-3).
- `MAX_VAL` specifies the upper limits (inclusive) for the four ISDC alert levels (DAL table columns 1-3 corresponds to alert levels 0-3).
- `SUB_ASSEMBLY` specifies the SPI PSD sub-assembly and must contain the entry `SPI_PSD`.

All other columns (`OBT_START`, `OBT_END`, `CHECK_MODE`, `ALERT_DELAY`) of the `SPI.-ALRT-LIM` structure are ignored. The validity of the alert limit file is defined by the two keywords `VSTART` and `VSTOP`. If an alert limit index is used, these keywords are used to selected the alert limit file that is appropriate for the PSD sensitivity improvement validity time interval. If the PSD sensitivity improvement validity time interval stops before the validity of the earliest alert limits, the earliest alert limits are used by `spi_psd_si` (a warning is issued by `spi_psd_si` in this case). If the validity time interval starts after the validity of the last alert limits, the last alert limits are used by `spi_psd_si` (a warning is issued by `spi_psd_si` in this case). If the PSD sensitivity improvement validity time interval overlaps with the transition of two (or more) alert limits files, those alert limits are applied that have the longest time overlap with the PSD sensitivity improvement validity time interval.

5 Algorithm

`spi_psd_si` compares two methods of analysing SPI data: one without considering PSD information, and one which takes into account the available PSD information.

If no PSD information is considered, all SPI events classified as PSD events `PE` become equivalent to single detector events `SE`, hence they are simply added together into a single spectrum. The gamma-ray line detection significance is then estimated from this spectrum using a maximum likelihood algorithm, which compares the two hypotheses:

H_1 : a gamma-ray line with unknown amplitude, energy, and width is present in the data

H_0 : no gamma-ray line is present in the data

(H_0 is often referred to as *null hypothesis*, H_1 is the *alternative hypothesis*). Following Cash[1], the log-likelihood ratio λ, defined as

$$\lambda = 2(\ln L_1 - \ln L_0), \tag{1}$$

is distributed like a χ^2_3 distribuion (3 degrees of freedom), where L_0 is the log-likelihood obtained using the null hypothesis, while L_1 is the log-likelihood obtained using the alternative hypothesis. This results then in a detection significance $S_{\texttt{ALL}}$.

If PSD information is considered, the SPI single detector events split into two classes:

- PSD multiple-site events (i.e. `PE` with a PSD flag of `1`)
- Single-detector events (`SE`) and PSD events that are not multiple-site events (i.e. `PE` with a PSD flag of `0` or for which an analysis error occured).

The gamma-ray line detection significance is then estimated for both spectra seperately, and subsequently added quadratically using

$$S_{\texttt{PSD}} = \sqrt{S^2_{\texttt{MUL}} + S^2_{\texttt{SGL}}} \tag{2}$$

The sensitivity improvement SI is then simply given by $SI = S_{\texttt{PSD}}/S_{\texttt{ALL}}$.

Optional alert limit checks are performed with higher alert levels preceeding lower alert levels, i.e. an alert of the highest possible level is generated. Minimum parameter limits are checked before maximum parameter limits, and minimum limits have precedence over maximum limits (this is not really of relevance since a parameter can not both violate the minimum and maximum limit **unless the alert limit file has not been set up correctly**, i.e. the minimum limit is always smaller or equal to the maximum limit). Alert limits are inclusive, i.e. a minimum limit violation alert is generated if

$$\texttt{PARAMETER} < \texttt{MIN\_VAL} \tag{3}$$

[1] Cash, W., 1979, Astrophysical Journal, 228, 939

is fulfilled, and a maximum limit violation alert is generated if

$$\texttt{PARAMETER} > \texttt{MAX\_VAL} \tag{4}$$

is fulfilled.

Limit violation alerts are only generated if

$$\texttt{PE\_NUM} \geq \texttt{minPE} \tag{5}$$

is fulfilled, where `PE_NUM` is the number of PSD events per detector in a time slice.

6 Potential gamma-ray lines

The following instrumental background lines have been identified so far in the PSD energy range:

Energy	Isotope	Single / Multiple site	Comments
205.31 keV	^{235}U	?	
570.0 keV	^{212}Po	?	identification uncertain
766.36 keV	^{234m}Pa	?	
1001.03 keV	^{234m}Pa	?	
1064.68 keV		?	no identification

7 Alerts

`spi_psd_si` may optionally generate alerts that signal possible PSD/SPI misfunctions. The following list provides the alert parameters and the actions that should be taken in case of occurence of the alerts. The alert parameter in the alert message is followed by the extension `_Ln` where *n=0-18* specifies the PSD detection channel for which the alert occured.

If the action **standard** is specified in the table, the standard alert action should be performed (**TBD**).

Parameter	Level	Action
`LINE_SIG_ALL`	0-3	standard
`LINE_SIG_MUL`	0-3	standard
`LINE_SIG_SGL`	0-3	standard
`PSD_SI`	0-3	standard

8 Error codes

The following error codes are defined (including `spi_toolslib` fit errors):

```
SPI_TOOLSLIB_ERROR_NO_CONVERGENCE              -231202   // spi_toolslib
SPI_TOOLSLIB_ERROR_FIT_STALLED                 -231203   // spi_toolslib
SPI_TOOLSLIB_ERROR_NR_SING_MATRIX              -231210   // spi_toolslib
SPI_PSD_SI_ERROR_MEM_ALLOC                     -231700
SPI_PSD_SI_ERROR_NO_LINES_REQUESTED            -231701
SPI_PSD_SI_ERROR_INVALID_SPI_MODE              -231702
SPI_PSD_SI_ERROR_NEGATIVE_ALL                  -231703
SPI_PSD_SI_ERROR_NEGATIVE_MUL                  -231704
```

```
SPI_PSD_SI_ERROR_NEGATIVE_SGL                -231705
SPI_PSD_SI_ERROR_ENERGY_NOT_FOUND            -231706
SPI_PSD_SI_ERROR_LINE_NOT_FOUND              -231707
SPI_PSD_SI_ERROR_INCOMPATIBLE_ENERGY         -231708
SPI_PSD_SI_ERROR_INDEX_SELECT                -231709
SPI_PSD_SI_ERROR_INVALID_LINE_DEF            -231710
```

They have the following meaning:

- `SPI_PSD_SI_ERROR_MEM_ALLOC` : the allocation of dynamical memory has failed. Probable your system resources are too limited to run this task.
- `SPI_PSD_SI_ERROR_NO_LINES_REQUESTED` : no gamma-ray line definitions have been found in task parameter file, hence no significance improvement estimation can be performed.
- `SPI_PSD_SI_ERROR_INVALID_SPI_MODE` : an invalid parameter has been specified for `spiMode`. Only `0` or `3` are allowed values. Correct task parameter file and restart task.
- `SPI_PSD_SI_ERROR_NEGATIVE_ALL` : the `SE + PE` spectrum has a negative content. This error should never occur. It may point towards a `spi_toolslib` error.
- `SPI_PSD_SI_ERROR_NEGATIVE_MUL` : the `PE` multiple-site spectrum has a negative content. This error should never occur. It may point towards a `spi_toolslib` error.
- `SPI_PSD_SI_ERROR_NEGATIVE_SGL` : the `PE` non-multiple-site spectrum has a negative content. This error should never occur. It may point towards a `spi_toolslib` error.
- `SPI_PSD_SI_ERROR_ENERGY_NOT_FOUND` : one of the specified gamma-ray lines has not been found in the spectra (check the log file to identify which line has not been found). Either the detection thereshold (defined by the task parameter `thres`) is set too high, or the specified line energy does indeed not correspond to a gamma-ray line that is detectible within the specified integration time. Set the task parameter `ignore = yes` to avoid `spi_psd_si` abortion in this case.
- `SPI_PSD_SI_ERROR_LINE_NOT_FOUND` : the specified gamma-ray line has not be found within the spectra. Set the task parameter `ignore = yes` to avoid `spi_psd_si` abortion in this case.
- `SPI_PSD_SI_ERROR_INCOMPATIBLE_ENERGY` : the best fitting line energies in the spectra `ALL`, `MUL` and `SGL` were incompatible. Set the task parameter `ignore = yes` to avoid `spi_psd_si` abortion in this case.
- `SPI_PSD_SI_ERROR_INDEX_SELECT` : while searching a single member in an index group, DAL3GEN returned more than one member. This should never happen. If this error occurs, it is likely that the alert limit index you specified on input is somehow corrupted.
- `SPI_PSD_SI_ERROR_INVALID_LINE_DEF` : an invalid line definition file has been found (not used so far).

In addition, the following `spi_toolslib` errors may signal fit problems:

- `SPI_TOOLSLIB_ERROR_NR_SING_MATRIX` : a fit encountered a singular matrix, hence `spi_psd_si` aborted. Set the task parameter `ignore = yes` to avoid `spi_psd_si` abortion in case of fit errors.
- `SPI_TOOLSLIB_ERROR_NO_CONVERGENCE` : a fit did not converge, hence `spi_psd_si` aborted. Set the task parameter `ignore = yes` to avoid `spi_psd_si` abortion in case of fit errors.
- `SPI_TOOLSLIB_ERROR_FIT_STALLED` : a fit stalled, hence `spi_psd_si` aborted. Set the task parameter `ignore = yes` to avoid `spi_psd_si` abortion in case of fit errors.